Spatial Filtering

Spatial Filtering

- Spatial filtering techniques take as input brain signals recorded from several different locations (or “channels”) and transform them in one of several ways.
- Possible goals include
  - enhancing local activity
  - reducing noise that is common across channels,
  - decreasing the dimensionality of the data,
  - finding projections that maximize discrimination between different classes

Bipolar

- Extract bipolar signals

$$\widetilde{s_{i,j}} = s_i - s_j$$

- Highlight the **electrical potential differences** between the two electrodes of interest (i and j).

Laplacian

- *Laplacian filtering*, extracts local activity at electrode *i* by subtracting the average activity present in the four orthogonal nearest neighboring electrodes

$$\tilde{\boldsymbol{s}} = \boldsymbol{s_i} - \frac{\mathbf{1}}{\mathbf{4}} \sum_{\boldsymbol{i} \in \boldsymbol{\theta}} \boldsymbol{s_i}$$

Common Average Referencing

- *Common average referencing* (CAR), enhances the local activity at electrode *i* by subtracting the average over all electrodes

$$\boldsymbol{\widetilde{s}_i = s_i - \frac{1}{N}\sum_{i=1}^{N} s_i}$$

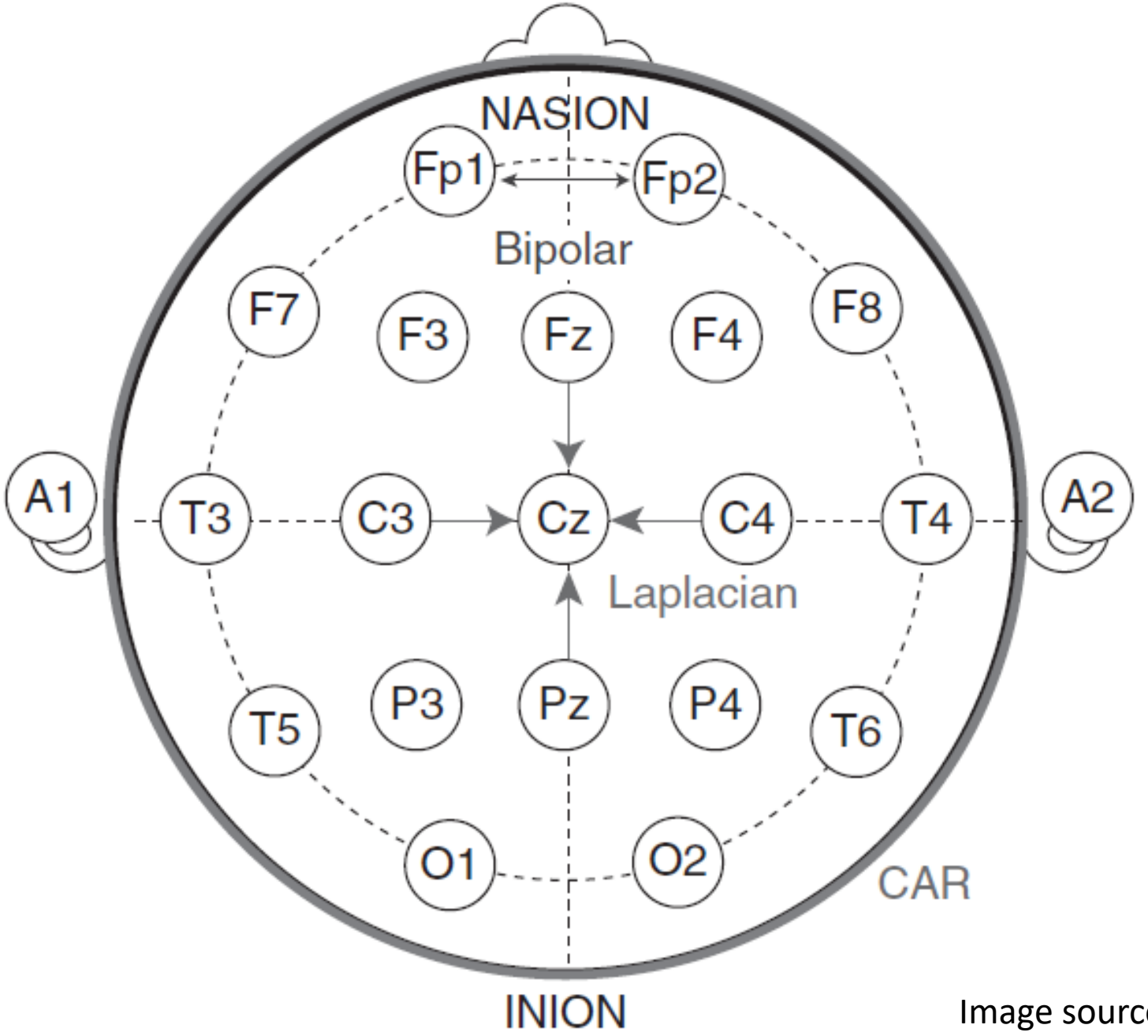


Image source: Rajesh P.N, Rao- Brain Computer Interfacing: An Introduction

Vector Representation

- A vector **x** ϵ R^n can be represented by n components:

$$\mathbf{x}: \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ x_N \end{bmatrix}$$

- Assuming the standard base <v_1, v_2, …, v_N> (i.e., unit vectors in each dimension), x_i can be obtained by projecting **x** along the direction of v_i:

$$x_i = \frac{\mathbf{x}^T v_i}{v_i^T v_i} = \mathbf{x}^T v_i$$

- **x** can be "reconstructed" from its projections as follows:

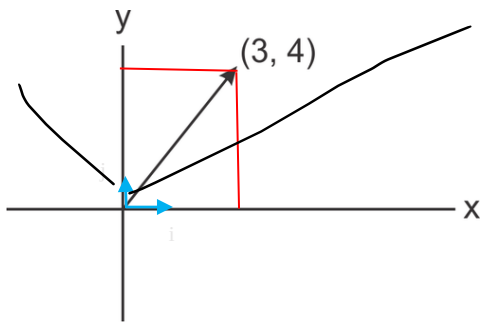
$$\mathbf{x} = \sum_{i=1}^{N} x_i v_i = x_1 v_1 + x_2 v_2 + \ldots + x_N v_N$$

- Since the basis vectors are the same for all x ϵ R^n (standard basis), we typically represent them as a n-component vector.

Vector Representation (cont'd)

- **Example** assuming n=2:

$$\mathbf{x}: \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$



- Assuming the standard base <v_1=i, v_2=j>, x_i can be obtained by projecting x along the direction of v_i:

$$x_1 = \mathbf{x}^T i = \begin{bmatrix} 3 & 4 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = 3$$

$$x_2 = \mathbf{x}^T j = \begin{bmatrix} 3 & 4 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 4$$

- **x** can be "reconstructed" from its projections as follows:

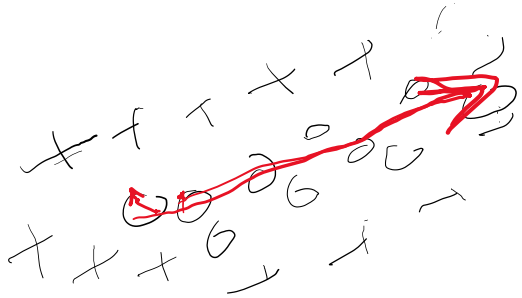
$$\mathbf{x} = 3i + 4j$$

Principal Component Analysis

- The goal in *principal component analysis* (PCA) (also called the *Karhunen-Loeve* or *Hotelling transform*) is to discover the underlying statistical variability in the data and reduce the data's dimensionality from D to a much smaller number of dimensions L ($L << D$).

- PCA achieves this goal by
  - Finding the directions of maximum variance in the D-dimensional data
  - Rotating the original coordinate system to align with these directions of maximum variance

Principal Component Analysis



- Most natural signals, including brain signals are redundant
- In the case of EEG measurements from N electrodes
  - Measurements from nearby electrodes may be correlated
  - Underlying rhythms across multiple electrodes.
- PCA attempts to find the dominant directions of variability in the data.
- New data points can be projected along the “principal” directions. Each projection is called a “principal component”
- The resulting *L*-dimensional vector can be used as a feature vector for classification or other purposes in BCI applications

Principal Component Analysis (PCA)

- If $\mathbf{x} \in R^N$, then it can be written a linear combination of an orthonormal set of N basis vectors $<v_1, v_2, \ldots, v_N>$ in R^N (e.g., using the standard base):

$$v_i^T v_j = \begin{cases} 1 & if \;\; i = j \\ 0 & otherwise \end{cases}$$

$$\mathbf{x} = \sum_{i=1}^{N} x_i v_i = x_1 v_1 + x_2 v_2 + \ldots + x_N v_N$$

$$where \quad x_i = \frac{\mathbf{x}^T v_i}{v_i^T v_i} = \mathbf{x}^T v_i$$

$$\mathbf{x}: \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ . \\ . \\ x_N \end{bmatrix}$$

- PCA seeks to approximate **x** in a subspace of R^N using a new set of K<<N basis vectors $<u_1, u_2, \ldots, u_K>$ in R^N:

$$\hat{\mathbf{x}} = \sum_{i=1}^{K} y_i u_i = y_1 u_1 + y_2 u_2 + \ldots + y_K u_K \quad where \quad y_i = \frac{\mathbf{x}^T u_i}{u_i^T u_i} = \mathbf{x}^T u_i$$

(reconstruction)

$$\hat{\mathbf{x}}: \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ y_K \end{bmatrix}$$

such that $\|\mathbf{x} - \hat{\mathbf{x}}\|$ is minimized!

(i.e., minimize information loss)

Principal Component Analysis (PCA)

- The "optimal" set of basis vectors $<u_1, u_2, \ldots, u_K>$ can be found as follows (we will see why):

(1) Find the eigenvectors u_i of the covariance matrix of the (training) data Σ_x

$$\Sigma_x u_i = \lambda_i u_i$$

(2) Choose the K "largest" eigenvectors u_i (i.e., corresponding to the K "largest" eigenvalues λ_i)

$<u_1, u_2, \ldots, u_K>$ correspond to the "optimal" basis!

We refer to the "largest" eigenvectors u_i as principal components.

PCA - Steps

- Suppose we are given $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M$ (N x 1) vectors

N: # of features

M: # data

Step 1: compute sample mean

$$\bar{\mathbf{x}} = \frac{1}{M}\sum_{i=1}^{M}\mathbf{x}_i$$

Step 2: subtract sample mean (i.e., center data at zero)

$$\Phi_i = \mathbf{x}_i - \bar{\mathbf{x}}$$

Step 3: compute the sample covariance matrix Σ_x

$$\Sigma_{\mathbf{x}} = \frac{1}{M}\sum_{i=1}^{M}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T = \frac{1}{M}\sum_{i=1}^{M}\Phi_i\Phi_i^T = \frac{1}{M}AA^T$$

where $A=[\Phi_1\ \Phi_2\ \ldots\ \Phi_M]$

i.e., the columns of A are the Φ_i

(N x M matrix)

PCA - Steps

Step 4: compute the eigenvalues/eigenvectors of Σ_x

$$\Sigma_x u_i = \lambda_i u_i$$

where we assume $\lambda_1 > \lambda_2 > \ldots > \lambda_N$

Note : most software packages return the eigenvalues (and corresponding eigenvectors) is decreasing order – if not, you can explicitly put them in this order)

Since Σ_x is symmetric, $<u_1,u_2,\ldots,u_N>$ form an orthogonal basis in R^N and we can represent any $\mathbf{x} \in R^N$ as:

$$\mathbf{x} - \bar{\mathbf{x}} = \sum_{i=1}^{N} y_i u_i = y_1 u_1 + y_2 u_2 + \ldots + y_N u_N$$

$$y_i = \frac{(\mathbf{x} - \bar{\mathbf{x}})^T u_i}{u_i^T u_i} = (\mathbf{x} - \bar{\mathbf{x}})^T u_i \qquad if \quad \| u_i \| = 1$$

i.e., this is just a "change" of basis!

$$\mathbf{x} - \bar{\mathbf{x}} : \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ x_N \end{bmatrix} \rightarrow \begin{bmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ y_N \end{bmatrix}$$

Note : most software packages normalize u_i to unit length to simplify calculations; if not, you can explicitly normalize them)

PCA - Steps

Step 5: dimensionality reduction step – approximate **x** using only the first K eigenvectors (K<<N) (i.e., corresponding to the K largest eigenvalues where K is a parameter):

$$\mathbf{x}-\bar{\mathbf{x}}=\sum_{i=1}^{N} y_i u_i = y_1 u_1 + y_2 u_2 + \ldots + y_N u_N$$

N → K

approximate $\mathbf{x}$ *by* $\hat{\mathbf{x}}$
using first K eigenvectors only

$$\hat{\mathbf{x}}-\bar{\mathbf{x}}=\sum_{i=1}^{K} y_i u_i = y_1 u_1 + y_2 u_2 + \ldots + y_K u_K$$

(reconstruction)

$\hat{x}$

$$\mathbf{x}-\bar{\mathbf{x}}:\begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ . \\ . \\ x_N \end{bmatrix} \rightarrow \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ . \\ . \\ y_N \end{bmatrix} \rightarrow \hat{\mathbf{x}}-\bar{\mathbf{x}}:\begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ y_K \end{bmatrix}$$

note that if K=N, then $\hat{\mathbf{x}}=\mathbf{x}$
(i.e., zero reconstruction error)

What is the Linear Transformation implied by PCA?

- The linear transformation y = Tx which performs the dimensionality reduction in PCA is:

$$\hat{\mathbf{x}} - \bar{\mathbf{x}} = \sum_{i=1}^{K} y_i u_i = y_1 u_1 + y_2 u_2 + \ldots + y_K u_K$$

$$(\hat{\mathbf{x}} - \bar{\mathbf{x}}) = U \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ y_K \end{bmatrix}$$

$$where \;\; U = [u_1 \, u_2 \ldots u_K] \;\; N \, x \, K \;\; \text{matrix}$$

i.e., the columns of U are the the first K eigenvectors of Σ_x

$$\Rightarrow \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ y_K \end{bmatrix} = U^T (\hat{\mathbf{x}} - \bar{\mathbf{x}})$$

$T = U^T$ K x N matrix

i.e., the rows of T are the first K eigenvectors of Σ_x

What is the form of Σ_y ?

$$\Sigma_{\mathbf{x}} = \frac{1}{M}\sum_{i=1}^{M}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T = \frac{1}{M}\sum_{i=1}^{M}\Phi_i\Phi_i^T$$

Using diagonalization:

$$\Sigma_{\mathbf{x}} = P\Lambda P^T$$

The columns of P are the eigenvectors of Σ_X

The diagonal elements of Λ are the eigenvalues of Σ_X or the variances

$$\mathbf{y}_i = U^T(\mathbf{x}_i - \bar{\mathbf{x}}) = P^T\Phi_i$$

$$\Sigma_{\mathbf{y}} = \frac{1}{M}\sum_{i=1}^{M}(\mathbf{y}_i - \bar{\mathbf{y}})(\mathbf{y}_i - \bar{\mathbf{y}})^T = \frac{1}{M}\sum_{i=1}^{M}(\mathbf{y}_i)(\mathbf{y}_i)^T = \frac{1}{M}\sum_{i=1}^{M}(P^T\Phi_i)(P^T\Phi_i)^T =$$

$$\frac{1}{M}\sum_{i=1}^{M}(P^T\Phi_i)(\Phi_i^T P) = P^T\left(\frac{1}{M}\sum_{i=1}^{M}\Phi_i\Phi_i^T\right)P = P^T\Sigma_{\mathbf{x}}P = P^T(P\Lambda P^T)P = \Lambda$$

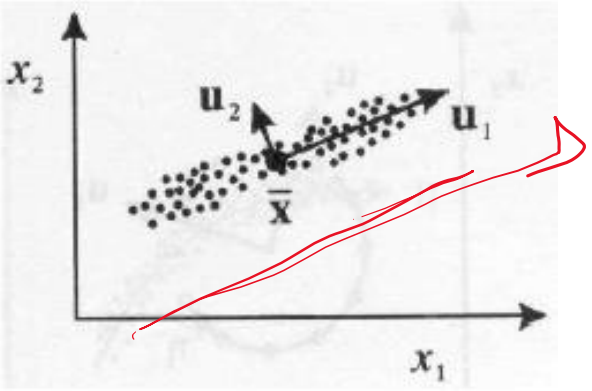
$$\boxed{\Sigma_{\mathbf{y}} = \Lambda}$$

PCA de-correlates the data!

Preserves original variances!

Interpretation of PCA

- PCA chooses the eigenvectors of the covariance matrix corresponding to the largest eigenvalues.
- The eigenvalues correspond to the variance of the data along the eigenvector directions.
- Therefore, PCA projects the data along the directions where the data varies most.
- PCA preserves as much information in the data by preserving as much variance in the data.



u_1: direction of max variance
u_2: orthogonal to u_1

Example

- Compute the PCA of the following dataset:

(1,2),(3,3),(3,5),(5,4),(5,6),(6,5),(8,7),(9,8)

- Compute the sample covariance matrix is:

$$\hat{\Sigma} = \frac{1}{n}\sum_{k=1}^{n}(\mathbf{x}_k - \hat{\boldsymbol{\mu}})(\mathbf{x}_k - \hat{\boldsymbol{\mu}})^t$$

$$\Sigma_x = \begin{bmatrix} 6.25 & 4.25 \\ 4.25 & 3.5 \end{bmatrix}$$

- The eigenvalues can be computed by finding the roots of the characteristic polynomial:

$$\Sigma_x v = \lambda v \Rightarrow |\Sigma_x - \lambda I| = 0$$
$$\Rightarrow \begin{vmatrix} 6.25 - \lambda & 4.25 \\ 4.25 & 3.5 - \lambda \end{vmatrix} = 0$$
$$\Rightarrow \boldsymbol{\lambda_1 = 9.34; \lambda_2 = 0.41}$$

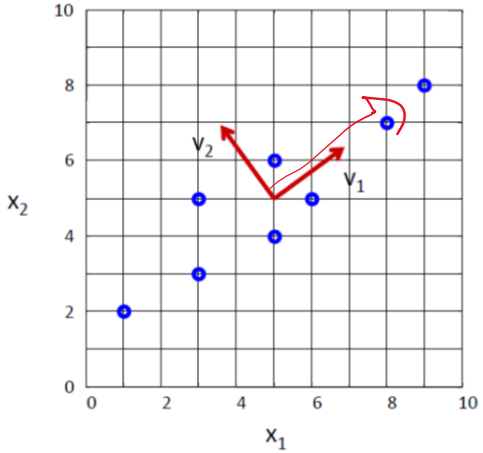
Example (cont'd)

- The eigenvectors are the solutions of the systems:

$$\Sigma_{\mathbf{x}} u_i = \lambda_i u_i$$

$$\begin{bmatrix} 6.25 & 4.25 \\ 4.25 & 3.5 \end{bmatrix} \begin{bmatrix} v_{11} \\ v_{12} \end{bmatrix} = \begin{bmatrix} \lambda_1 v_{11} \\ \lambda_1 v_{12} \end{bmatrix} \Rightarrow \begin{bmatrix} v_{11} \\ v_{12} \end{bmatrix} = \begin{bmatrix} 0.81 \\ 0.59 \end{bmatrix}$$

$$\begin{bmatrix} 6.25 & 4.25 \\ 4.25 & 3.5 \end{bmatrix} \begin{bmatrix} v_{21} \\ v_{22} \end{bmatrix} = \begin{bmatrix} \lambda_2 v_{21} \\ \lambda_2 v_{22} \end{bmatrix} \Rightarrow \begin{bmatrix} v_{21} \\ v_{22} \end{bmatrix} = \begin{bmatrix} -0.59 \\ 0.81 \end{bmatrix}$$



Note: if u_i is a solution, then cu_i is also a solution where $c \neq 0$.

Eigenvectors can be normalized to unit-length using:

$$\hat{v}_i = \frac{v_i}{\| v_i \|}$$

How do we choose K ?

- *K* is typically chosen based on how much information (variance) we want to preserve:

Choose the smallest K that satisfies the following inequality:

$$\frac{\sum_{i=1}^{K} \lambda_i}{\sum_{i=1}^{N} \lambda_i} > T \quad \textit{where } T \textit{ is a threshold } (e.g., 0.9)$$

- If T=0.9, for example, we "preserve" 90% of the information (variance) in the data.

- If K=N, then we "preserve" 100% of the information in the data (i.e., just a "change" of basis and $\hat{\mathbf{x}} = \mathbf{x}$)

Approximation Error

- The approximation error (or reconstruction error) can be computed by:

$$\| \mathbf{x} - \hat{\mathbf{x}} \|$$

where $\hat{\mathbf{x}} = \sum_{i=1}^{K} y_i u_i + \bar{\mathbf{x}} = y_1 u_1 + y_2 u_2 + \ldots + y_K u_K + \bar{\mathbf{x}}$

(reconstruction)

- It can also be shown that the approximation error can be computed as follows:

$$\| \mathbf{x} - \hat{\mathbf{x}} \| = \frac{1}{2} \sum_{i=K+1}^{N} \lambda_i$$

Data Normalization

- The principal components are dependent on the *units* used to measure the original variables as well as on the *range* of values they assume.

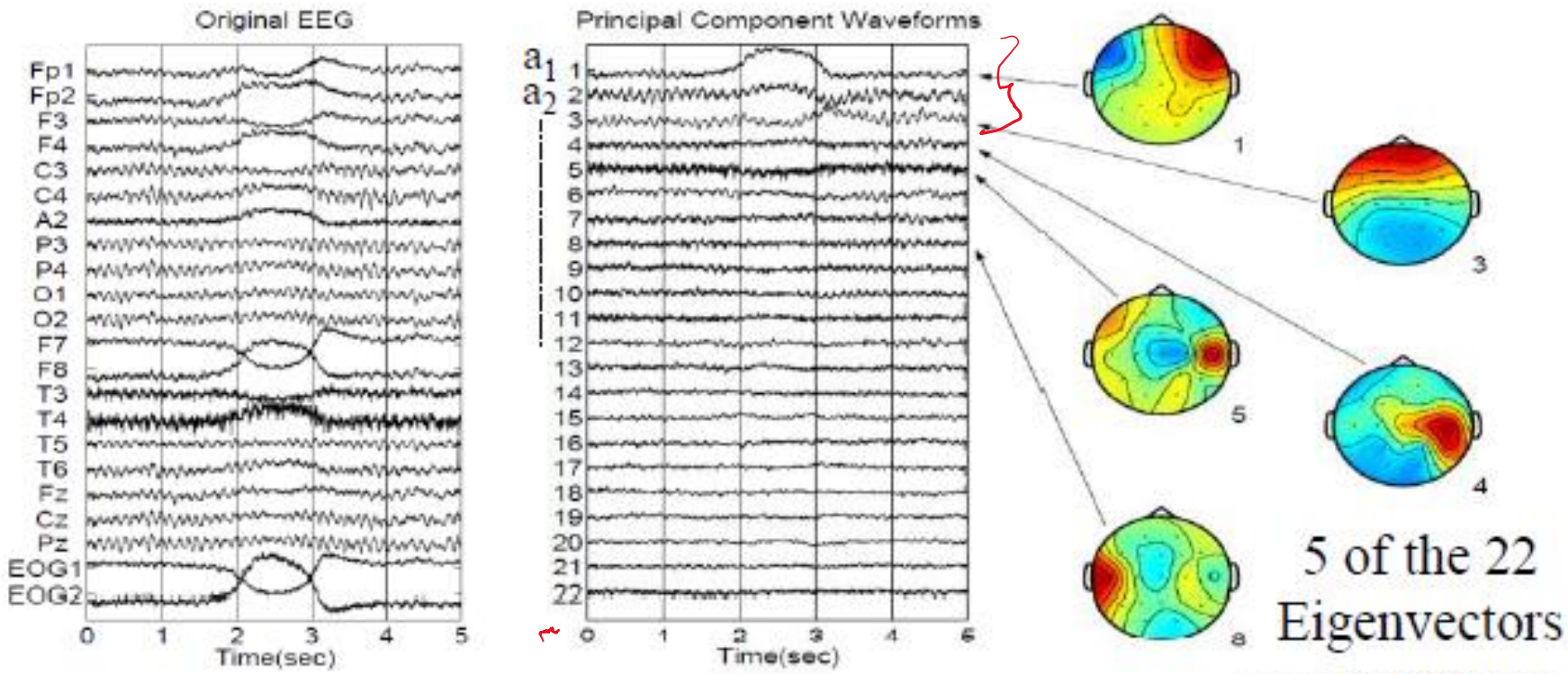
- Data should always be normalized prior to using PCA.

- A common normalization method is to transform all the data to have zero mean and unit standard deviation:

$$\frac{x_i - \mu}{\sigma}$$

where μ and σ are the mean and standard deviation of the i-th feature x_i

PCA applied to EEG



5 of the 22 Eigenvectors (spatial filters for EEG channels)

A

B.

(Jung et al., 1998)

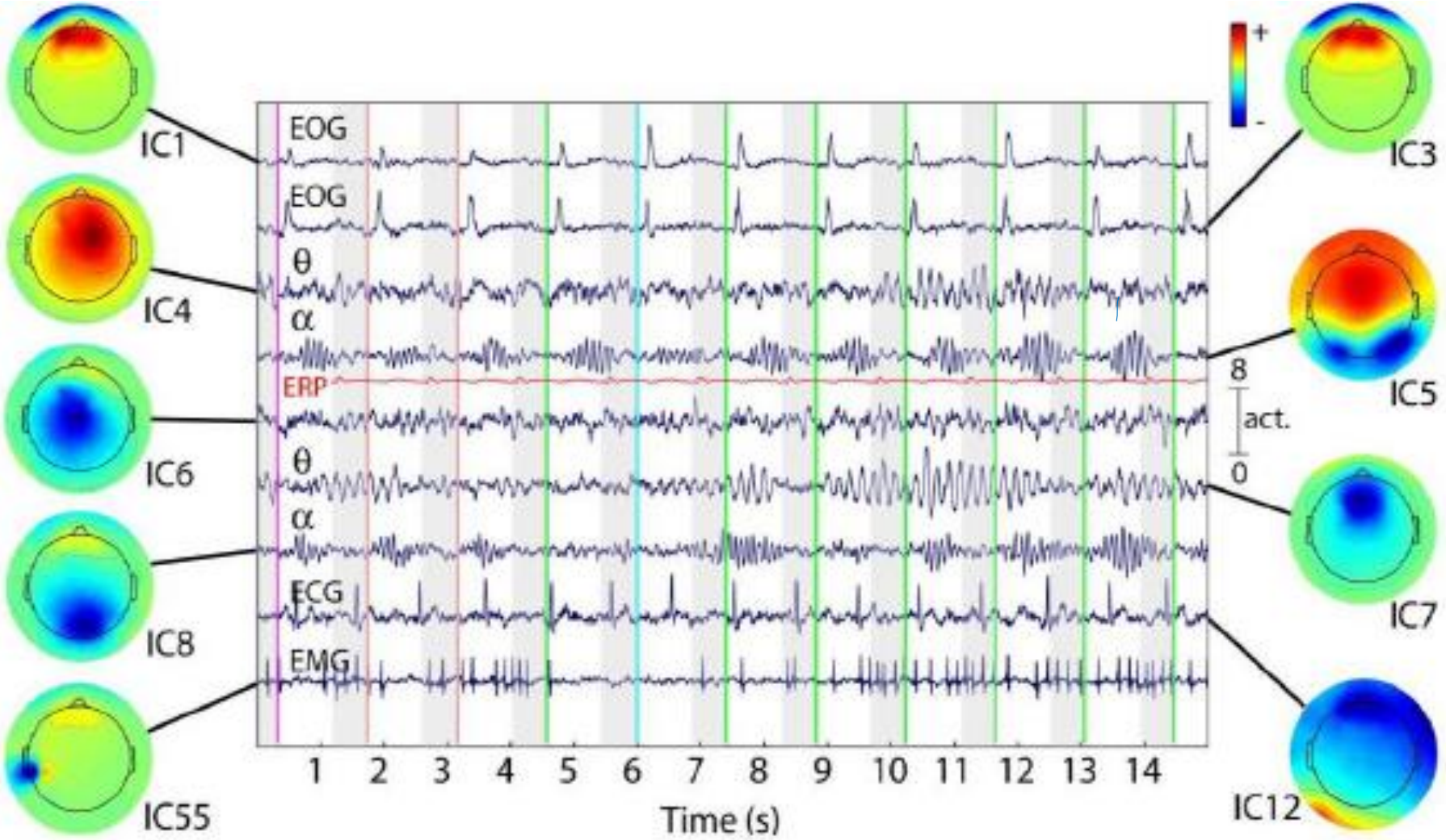
Independent Component Analysis

- PCA finds a matrix **V** that decorrelates the inputs but the resulting feature vector ***a*** may still retain higher order statistical dependencies
- There may be a possibility that the variables are independent.
- ICA tries to find a matrix W of filters (columns of W) such that the output **a** is **statistically independent**:

$$\boldsymbol{a} = \boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{x} \text{ such that } \boldsymbol{P}(\boldsymbol{a}) \approx \prod_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{D}} \boldsymbol{P}(\boldsymbol{a}_{\boldsymbol{i}})$$

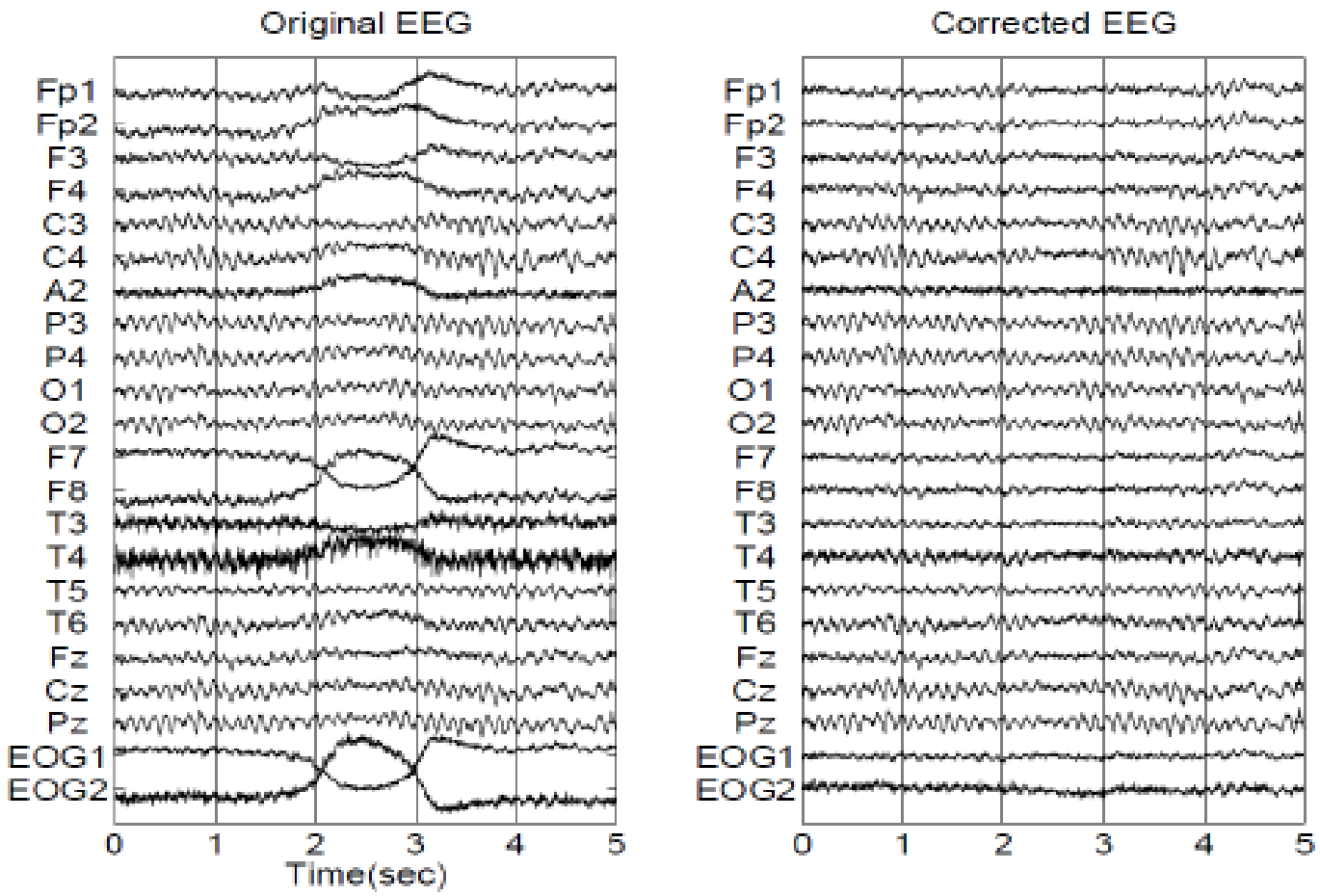
Independent Component Analysis

- ICA assumes sources are linearly mixed to produce x
- The feature vector dimension in ICA can be lesser than, equal to, or greater than the number of input dimensions.
- ICA has proved useful in a variety of settings in BCI applications, ranging from the use of the output vector **a** as a feature vector in classification.



Application of ICA to EEG data for isolating electro- oculographic (EOG) (eye-related), electromyographic (EMG) (muscle-related) and electrocardiographic (ECG) (heart-related) artifacts, and unmixing putative source signals in the brain. Image (adapted from Onton and Makeig, 2006)

ICA for Artifact Removal in EEG



Common Spatial Pattern

- Supervised Technique
- Data is labeled with class to which each data vector belongs
  - E.g., EEG obtained for right versus left hand imagery
- CSP finds a matrix of spatial filters
  - the variance of the filtered data for one class is maximized
  - variance of the filtered data for the other class is minimized
- CSP filters can significantly enhance discrimination ability between the two classes

Common Spatial Pattern

$$\begin{array}{c|ccc} & ch\text{-}1 & ch\text{-}2 & ch\text{-}3 \\ \hline 1 & 2.5 & 3.5 & 4.5 \\ 2 & 0.3 & 2.8 & 1.9 \\ 3 & \vdots & \vdots & \vdots \end{array} \quad N \times T$$

<u>Input</u> : $\{X_c^i\}_{i=1}^{K}$ c: classes, i: trial, K: no. of trials.

$X_c^i \rightarrow N \times T$ [No. of channels × Time samples]

X - Matrix

x - vector

Assuming X_c^i is centered & scaled

<u>Goal</u>:- Spatial filter Matrix 'W' $\rightarrow N \times M$

Tranform signal according to eqn.

$$x_{csp}(t) = \underline{\underline{W^T}}\, x(t) \quad \text{—— (1)}$$

Common Spatial Pattern

Two class conditional covariance matrix

$$R_c = \frac{1}{K} \sum_i X_c^i (X_c^i)^T \quad \text{for } c \in [1,2]$$

determing W as

$$W^T R_1 W = \Lambda_1$$

$$W^T R_2 W = \Lambda_2$$

$$\begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_1 \end{bmatrix}$$

$$\Lambda_1 + \Lambda_2 = I$$

$$R_1 W = \lambda R_2 W$$

Generalized eigenvalue

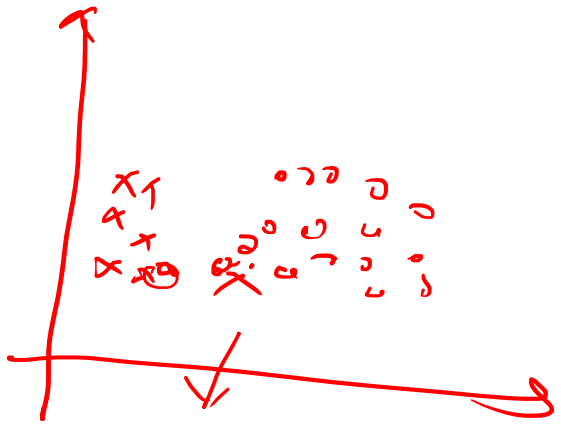
$$\lambda_1^j = w_j^T R_1 w_j \rightarrow \lambda_1$$

$$\lambda_2^j = w_j^T R_2 w_j \rightarrow \lambda_2$$

$$\lambda_1^j + \lambda_2^j = 1$$

High value
High variance

Low value
Low variance



Classifier

CSP applied to EEG for Right/Left Hand Imagery

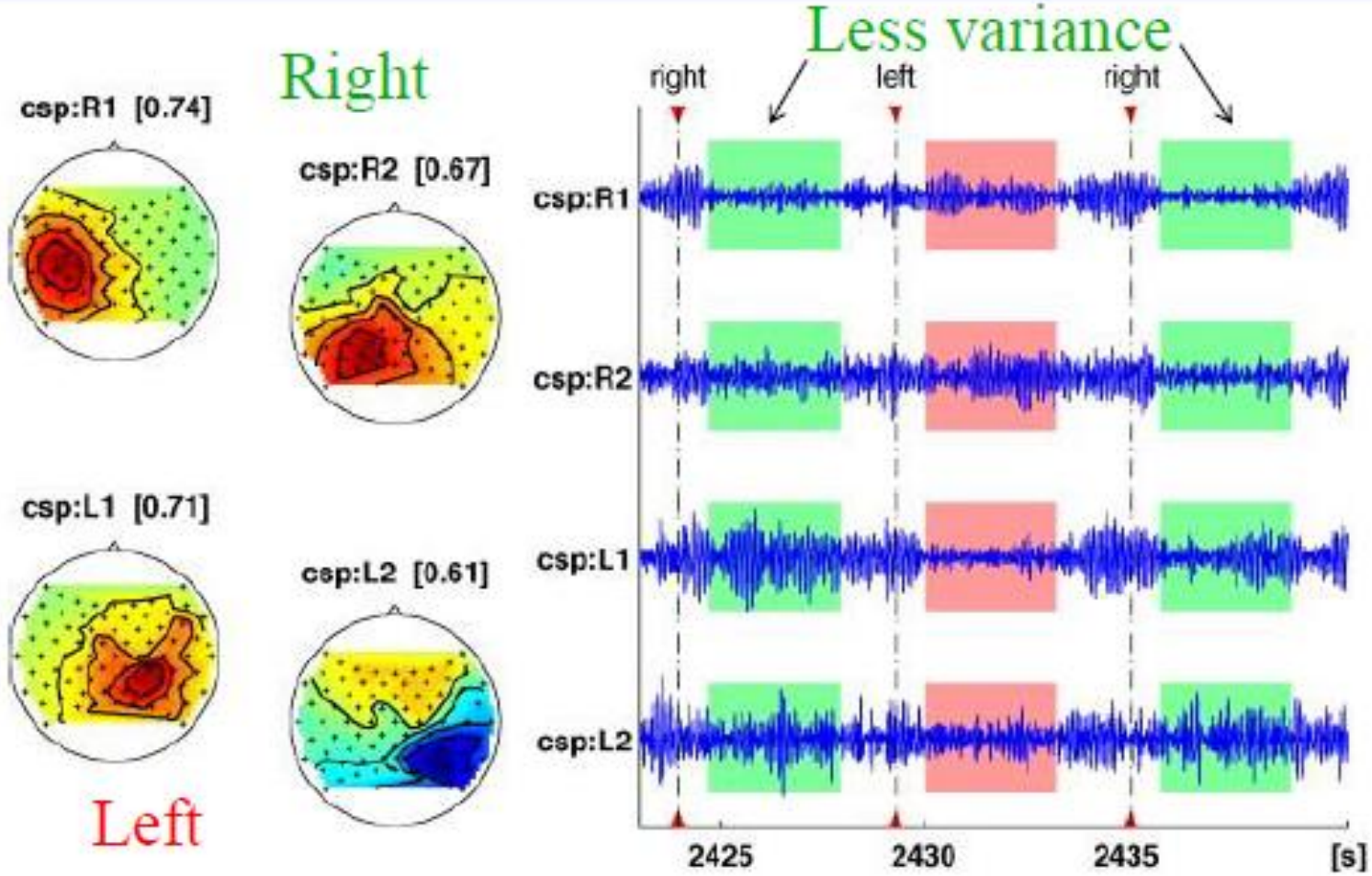


Image source: http://www.sciencedirect.com/science/article/pii/S0165027007004657

Artifact Reduction Techniques

- Artifacts in BCIs are any undesirable signals
  - Artifacts outside body-50/60Hz
    - Power line
    - External electrical interference
  - Artifacts within body
    - Rhythmic artifacts due to respiration and heartbeat (the latter are called electrocardiographic or ECG artifacts
    - Signal distortion or attenuation due to skin conductance changes
    - Eye movement and eye blink artifacts (also called electro-oculographic or EOG artifacts)-- range 3–4Hz
    - Muscle artifacts (electromyographic or EMG artifacts)-- 30Hz or higher frequency range.

Artifact Reduction Techniques

- Thresholding
  - If the magnitude or some other characteristic of a recorded EOG or EMG signal exceeds a pre-determined threshold, the brain signals recorded during that epoch are deemed to be contaminated and rejected.
- Band-Stop and Notch Filtering
  - Band-stop filtering is a useful artifact reduction technique that attenuates the components of a signal in a specific frequency band and passes the rest of the components of the signal.
  - A notch filter set to the 59–61 Hz band (in the United States) for filtering out the 60 Hz power-line noise artifact.

Artifact Reduction Techniques

- Linear Modeling
  - A simple way of modeling the effect of artifacts on a recorded brain signal is to assume that the effect is additive.
  - For example, if $EEG_i(t)$ is the EEG signal recorded from electrode *i* at time *t*, then a model of how the signal has been contaminated could be:

$$EEG_i(t) = EEG_i^{true}(t) + K \cdot EOG(t)$$

  - $EEG_i^{true}(t)$ is the uncontaminated ("true") EEG signal from electrode *i* at time *t*, $EOG(t)$ is the recorded EOG signal at time *t* and *K* is a constant.
  - Given an estimated value for K, one can obtain an estimate of the true EEG signal using:

$$EEG_i^{true}(t) = EEG_i(t) - K \cdot EOG(t)$$